Preface

Negotiation mechanisms have been studied widely in the field of multi-agent systems. They possess a variety of features that enable agents be negotiate with each other even in open environments. However, mainly because of limited computational power, there are several assumptions that traditionally limit the degree of openness. Recent studies have tended to focus on completely open and highly uncertain environments that apply agent systems to the real world. For example, in emergency rescue domains, we cannot expect to know when and where a fire starts and when humans are likely to be injured. Also, in Internet auctions, there can be shill bids since there are many unauthenticated participants. Nowadays, we can employ machines with large computational power to compute an optimal way for agents to negotiate, even in completely open and highly uncertain environments. For the practical use of multi-agent systems in the real world, the reliability of each agent's behavior is essentially required. Concretely, agents must obtain the most appropriate solution/solutions based on rational, robust, and secure negotiation among multiple agents even if the environment is intractable. We solicit papers on all aspects of such negotiation mechanisms in multi-agent systems, including multi-issue negotiations, concurrent negotiations, strategy-proof mechanisms, rational argumentation, auctions and voting, and so on. These issues are being explored by researchers from different communities in multi-agent systems. They are, for instance, being studied in agent negotiation, multi-issue negotiations, auctions, mechanism design, electronic commerce, voting, secure protocols, matchmaking and brokering, argumentation, and co-operation mechanisms.

Nagoya, Japan *Takayuki Ito*
Kyoto, Japan *Hiromitsu Hattori*
Wollongong, Australia *Minjie Zhang*
Yamagata, Japan *Tokuro Matsuo*
January 2008

A Multi-Issue Negotiation Protocol Among Nonlinear Utility Agents: A Preliminary Report

Takayuki Ito[1], Mark Klein[2], and Hiromitsu Hattori[3]

[1] Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology, Gokiso, Showa-ku, Nagoya 466-8555, Japan ito.takayuki@nitech.ac.jp

[2] Center for Collective Intelligence, Sloan School of Management, Massachusetts Institute of Technology, Three Cambridge Center, NE20-336, Cambridge, MA 02142, USA m_klein@mit.edu

[3] Department of Social Informatics, Graduate School of Informatics, Kyoto University, Yoshida-Honmachi Sakyo-ku, Kyoto 606-8501, Japan hatto@i.kyoto-u.ac.jp

1 Introduction

Multi-issue negotiation protocols have been studied very widely and represent a promising field since most of negotiation problems in the real-world are complex ones including multiple issues. In particular, in reality, issues are constrained each other. This makes agents' utilities nonlinear. Further, even in collaborative situation, to get an agreement, agents need to act competitively because of their self-interested nature.

For example, when two designers collaboratively design a new car, there are multiple issues, e.g., color, engine, style, etc. They have preference over each issue, and there are constraints between the issues as well. For example, if the size of tires is large and the body style is R.V., then the size of the engine needs to be larger than 2,500 cc. This kind of interdependency between issues is ubiquitous in the real-world. The interdependency among issues makes agents' utilities very complex. This complex utility eventually can not be modeled as a simple linear utility function. We have to model such complex utility as completely non-linear utility function. In addition, a constraint between the style and the size of the engine can be different between designer's companies. Because these companies often hope to use their own parts for a new car, the designers are now in a competitive situation. Agents thus need to compete to get a desirable agreement over constraints as well as over issue values.

We propose an auction-based multiple-issue negotiation protocol among nonlinear utility agents. In order to make the protocol scalable, we first employ a sampling method for agents. By sampling its own utility space, an agent

T. Ito et al.: *A Multi-Issue Negotiation Protocol Among Nonlinear Utility Agents: A Preliminary Report*, Studies in Computational Intelligence (SCI) **89**, 25–38 (2008)
www.springerlink.com

can reduce its search cost. Also, the simple sampling often fails to find better solutions. Thus, in our protocol, agents adjust their sampled points firstly by using a search technique. After that, agents submit bids. Since we assume a huge utility space, if these bids are based on contract points, there could be too much bids. Thus, in our model, agents make bids on a set of constraints among issue values. This bid expression can drastically reduce the computational cost. The mediator finds a combination of bids that maximizes social welfare. Our experimental results show that our method can outperform the existing simple methods in particular in the huge utility space that can be often found in the real-world. Further, theoretically, our negotiation protocol can guarantee to find the optimal point if the sampling rate is sufficiently small and the threshold for selecting bids is 0.

There are a lot of previous works on multi-issue negotiation [1–6]. These efforts differ from our work since our protocol is attacking against handle completely nonlinear utilities. Most existing work also assumes that agents are totally collaborative or have linear utility functions. Our work focuses on mainly competitive agents and nonlinear utility functions. The details are shown in Sect. 5.

The rest of the paper is organized as follows. First we describe a model of nonlinear utility multi-issue negotiations. Here we define the nonlinear utility function. Second we propose a bargaining protocol that achieves a desirable solution in nonlinear utility multiple issue negotiations. Here, we propose an auction based bargaining protocol and a heuristic method for speeding up the protocol. Third we demonstrate the experimental results. Then, we compared our work with the existing work to clarify the features of our method, and concluding remarks are given in the final section.

2 A Negotiation Model Based on Nonlinear Utility

2.1 The Model

We consider the following situation with n agents who want to reach an agreement. An agent is represented by $a_i \in N$. There are m issues, $s_j \in S$, for negotiation. The number of issues represents the number of dimensions of the utility space. For example, if there are 3 issues, the utility space becomes 3 dimensional spaces. An issue s_j has a value, $[0, X]$, i.e., $s_j \in [0, X]$. There are l constraints, $c_k \in C$. A constraints represents a hyper dimensional solid among multiple issues. Figure 1 shows an example of a constraint between issue 1 and issue 2. This constraint has value of 55, and hold if the issue values for issue 1 are $[3, 7]$ and the issue values for issue 2 are $[4, 6]$.

A contract is represented by a vector $\mathbf{s} = (s_1, \dots s_m)$. Agent a_i has value $v_{a_i}(c_k, \mathbf{s})$ on a constraint c_k with a contract $\mathbf{s}$. $v_{a_i}(c_k, \mathbf{s})$ has a positive value if constraint c_k is satisfied on contract $\mathbf{s}$. In the real-world, $v_{a_i}(c_k, \mathbf{s})$ varies very much among different contacts and different constraints. This makes agent's utility space intractably nonlinear.

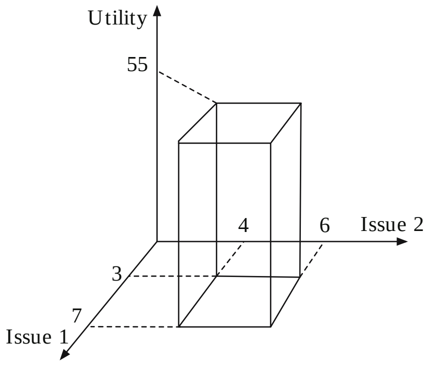

Fig. 1. Example of a constraint

2.2 Nonlinear Utility

Figure 2 shows an example of a nonlinear utility space. There are 2 issues, i.e., 2 dimensions and $X = 100$ for each issue. Also, there are 50 constraints that related to 1 issue and 100 constraints that related to 2 issues. The utility space is completely bumpy and there are many hills and valleys.

If we use a linear expression, agent's utility is defined as follows: $u_{a_i}(\mathbf{s}) = \sum_{c_k \in C} v_{a_i}(c_k, \mathbf{s})$. This expression looks linear. However, agent's utility space is nonlinear in the sense that the utility does not have a linear expression against contract $\mathbf{s}$. The interdependency among issues, which is represented as a constraint c_k, makes the utility space non-linear in terms of contracts. This is because the utility of higher dimensional constraints that depend on multiple issues can not be expressed by a linear function on a single issue. This point differs very much from the other existing works in which any dependency among issues are not assumed. Therefore, in our model, an utility space has a totally bumpy shape, which can not be represented a usual functional representation.

Another important point is that $v_{a_i}(c_k, \mathbf{s})$ can not be known from the other agents. Even agent a_i does not know the value when he calculates the value. This means that in the model agents are situated under an uncertain environment. Our protocol can be employed for such an uncertain environment.

On the contrary, there could be a simple nonlinear utility function that, for example, can be defined as like $u_i = f(s_1) * g^2(s_2)$. This function is nonlinear. However, this kind of nonlinear function constructs a simple shape utility space in which the optimal contract is a single or optimal contracts can be easily calculated from utility functions and the contracts.

Finding an optimal contract for a single agent in the utility space such as Fig. 2 is actually a multi objective optimization problem. Simulated annealing and evolutionary algorithms have been developed in the AI field and OR field for such optimization problem. However, we consider negotiation among two or more agents. Agents do not want to reveal their preference very

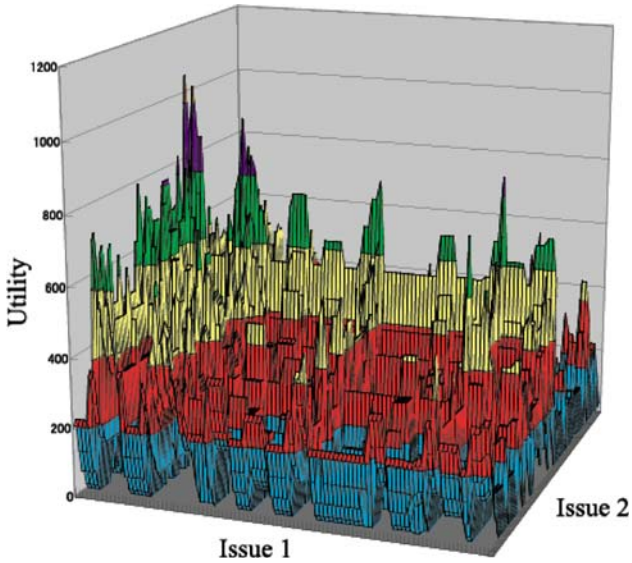

Fig. 2. Example of nonlinear utility space for a single agent

much. Thus, we can not just employ such methods, i.e., simulated annealing and evolutionary algorithms, because such methods assume to reveal such preferences.

2.3 Finding Pareto Efficient Contracts

The objective function for our protocol can be described as follows:

$$\arg\max_{\mathbf{s}} \sum_{a_i \in N} u_{a_i}(\mathbf{s}) \tag{1}$$

Namely, our protocol tries to find a contract point that maximizes social welfare, i.e., the total utilities of agents. Such a contract point eventually satisfies Pareto Efficiency.

If we use an exhaustive search, when there are M issues and X values for each issue, the utility space becomes X^M. This space is actually intractable when the size M and the size X become large. Thus, in our protocol, we propose to employ a sampling method for sampling such a huge utility space. There can be a case in which sampling fails to get accurate contract points. Thus we also propose to employ adjusting method for sampling. Namely, in our protocol, after sampling some points, an agent conduct simple searches from each point. This method perform very well for huge utility spaces.

3 Auction-Based Negotiation Among Agents

Our auction-based negotiation protocol is defined by the following four steps.

Step 1: Sampling Each agent gets samples in its utility space. The sampling rate α is defined by the protocol designer or the mediator. Figure 3 shows this concept. If the sampling rate α is not adequate, it often fails to get adequate contract points as sampling points.

Step 2: Adjusting Each agent adjusts samples by using a simulated annealing method. This step helps to adjust the sampling point. Only sampling often fails to get more feasible contracts without this step. From each sampled contract point, an agent conducts a simulated annealing method. In fact, this conducts multiple simulated annealing in the utility space. Also, this step make it possible to increase the sampling rate at step 1. Figure 4 shows this concept in ideal situation. By simulated annealing each sampling point may move to its close optimal contract point.

Step 3: Bidding Each agent make bids. For each sampled contract points, an agent valuates its utility. If the utility is larger than the threshold δ, then he packs a set of constraints into a single bid. The bid value is the value of the contract point which is a sum of values of constraints included in the bid. The threshold δ is defined by the protocol designer or the mediator. Figure 5 shows this concept.

Step 4: Maximizing Social Welfare The mediator finds combinations of bids that shares at least some of contract points (consistency) and maximize the total value of the bids (maximization). In this step, the mediator can employ a breadth-first search with branch cutting based on the above consistency. The size of the search space of the mediator depends on the number of constraints. The number of constraints can be much less than the number of the contract points. Thus, this constraint-based finding mechanism for the mediator can

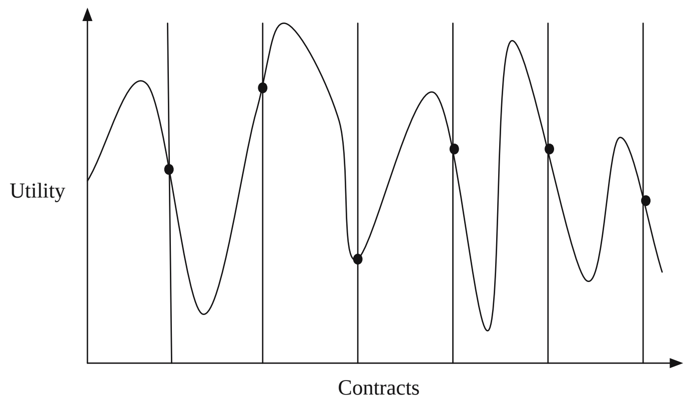

Fig. 3. Sampling utility space

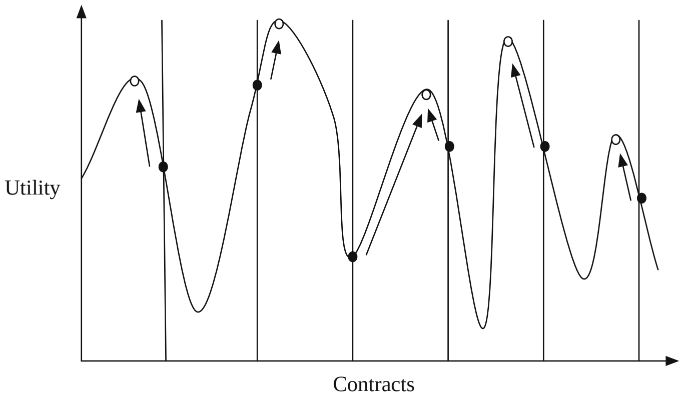

Fig. 4. Adjusting sampled contract points

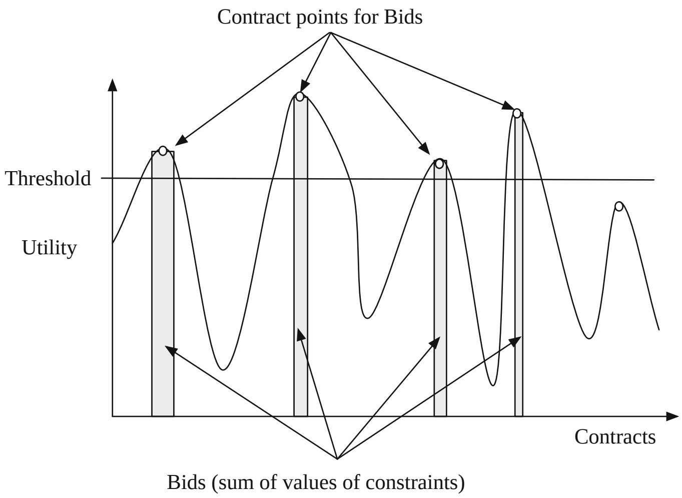

Fig. 5. Making bids

reduce the computational cost very much compared with an exhaustive search. Figure 6 shows this concept.

It is clear that we have the following proposition on the completeness.

Proposition 1 (Completeness). *If the threshold δ is 0 and the sampling rate α is 1, the proposed method can achieve the optimal point.*

Proof. If the threshold δ is 0, then the agent submits all possible bids on the sampled contract points. If the sampling rate α is 1, then the agent searches all possible contracts. Therefore, if $\delta = 0$ and $\alpha = 1$, then the agent submits all possible bids on the all possible contracts. Thus, the mediator searches all

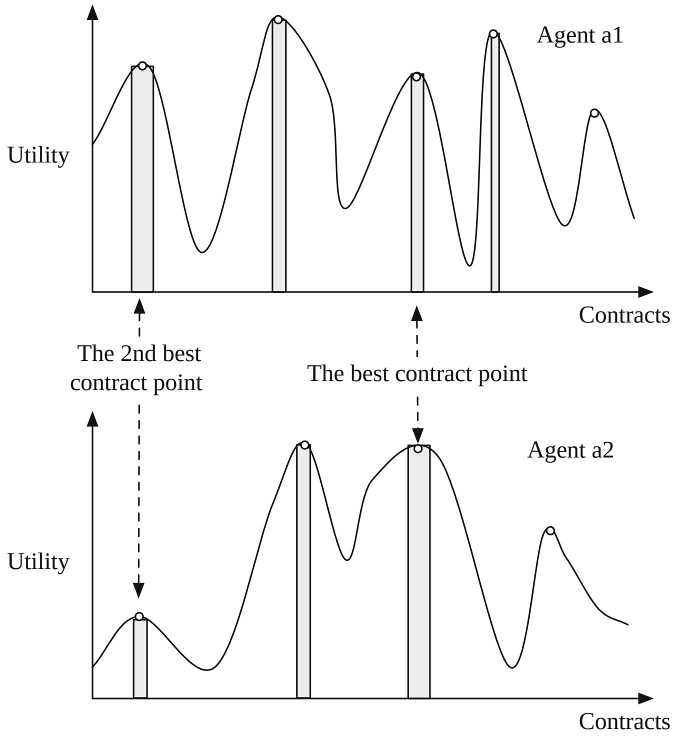

Fig. 6. Maximizing social welfare

possible combinations of the submitted bids that maximizes social welfare, i.e., the sum of utilities among agents. This process is exactly same as an exhaustive search in which the mediator searches the contract points that maximizes the sum of utilities among agents.

In fact, the completeness and the computational cost are a trade-off relation. Thus, we have to carefully adjust the threshold and the sampling rate based on the figure of utility spaces.

4 Preliminary Experiments

4.1 Setting

We conducted several experimentations to show the effectiveness and scalability of our approach.

We compared our approach with the other several approaches. Concretely, we constructed two search methods and three negotiation methods. The two search methods include an exhaustive search and a simulated annealing search.

The three negotiation methods include the local search-based negotiation, the negotiation method with simple sampling, and the negotiation method with SA-sampling.

The search methods basically find Pareto efficient points in order to evaluate our proposed methods. The exhaustive search method tries to search all possible contracts. The simulated annealing search method employs a simulated annealing search [7] in which the initial temperature is 50.0, decreasing the temperature 1/50 for each step, and stop if the temperature reaches at 0. The initial contract point is randomly selected. The important point is that the search methods have the entire utility spaces that aggregates all utilities among agents. On the contrary, the negotiation methods do not assume to have such an entire information since we assume agents reveal their information as less as possible.

The local search-based negotiation in the negotiation methods employs a local search mediator who starts from a random contract point. Then, the local search mediator randomly select a next candidate point from its neighbors. If all agents can agree to move to the next candidate point, then the mediator moves to the next point. Each agent makes an agreement if the next contract has higher value than the previously accepted agreement for him/her. This method obviously tend to stuck into local optimal points. The negotiation method with simple sampling is the negotiation method that does not use the step 2, in which sampling points are adjusted by simulated annealing. This method often fails to find adequate sampling points. The negotiation method with SA-sampling is the method we proposed in the previous section.

Out negotiation methods has a lot of parameters. In this experiments, we aim to show the scalability of our negotiation method with SA-sampling with respect to efficiency. Thus we set the parameters as follows:

- Number of agents: 3.
- Number of issues: 1 to 10.
- Number of constraints for each dimension: 5. This means that the number of the constraints that related on 1 issues is 5, 2 issues is 5, ..., and 10 issues is 5.
- The domain of the issue value is $[0, 10]$.
- The maximum value for a constraint is 100.
- The maximum range for feasible issue value is 7. This means that there may be a constraints that is hold under the issue values from 3 to 10.
- The sampling rate for negotiation methods: 10. This mean that for one dimension the method samples a single point. This is fairly large sampling rate since the maximum issue value is 10.
- The threshold for making bids in the negotiation methods: 100.

This parameter setting is one of many possibilities. For example, to show the scalability of the negotiation method with SA-sampling, we can set the sampling rate and the threshold more carefully.

4.2 Results

Figure 7 shows the actual social welfare when varying the number of issues. As you can see in the figure, the exhaustive search terminated after 7 issues because of its high computational cost. The negotiation method with SA-sampling outperforms the other negotiation methods. Further, interestingly, even the negotiation method with SA-sampling does not have the entire information of all agents' utility space, this method outperforms SA search method when the number of issues is large. This is more clear in Fig. 8.

In Fig. 8, we show the optimality rate compared with simulated annealing as an efficiency criterion. The exhaustive search should be such a criterion. However, the exhaustive search can not be employed in the huge utility spaces.

Impressively, when the number of issues is larger than 4, the negotiation method with SA-sampling outperform the simulated annealing search. The reason can be described as follows: When the utility space is huge, the simulated annealing search often fails into local optimal. However, since the negotiation method with SA-sampling can have several points to start with, the risk to lose optimal points is lower than a single simulated annealer even if the single simulate annealer has the global search space among agents.

In the utility space is small, the negotiation method with SA-sampling is lower than the centerized simulated annealing. Actually, we could expect this fact. When the utility space is small, both methods can find optimal points. However, the negotiation method does not know the entire optimality. In the other words, in the negotiation method, although each agent can get the private optimal point by each simulated annealing-based sampling, such the private optimal points does not necessarily the global optimal points.

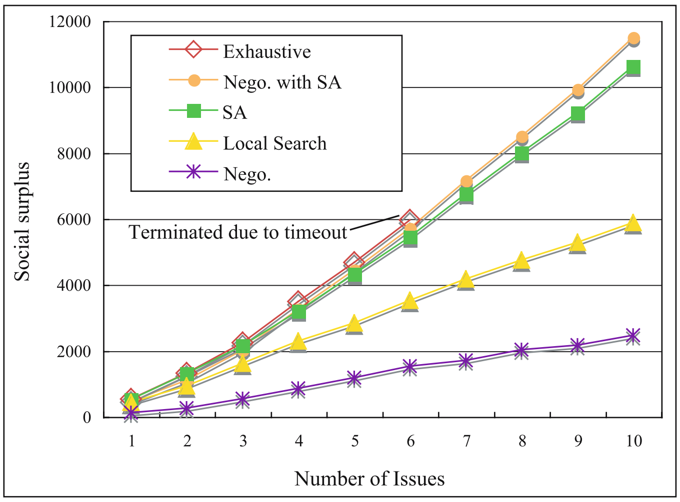

Fig. 7. Result 1: Social welfare

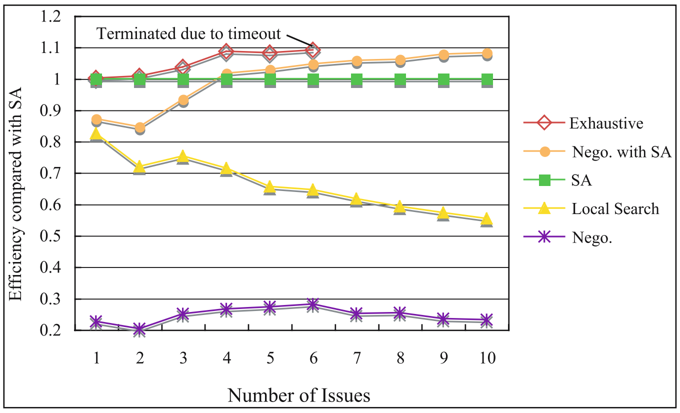

Fig. 8. Result 2: Efficiency compared with simulated annealing

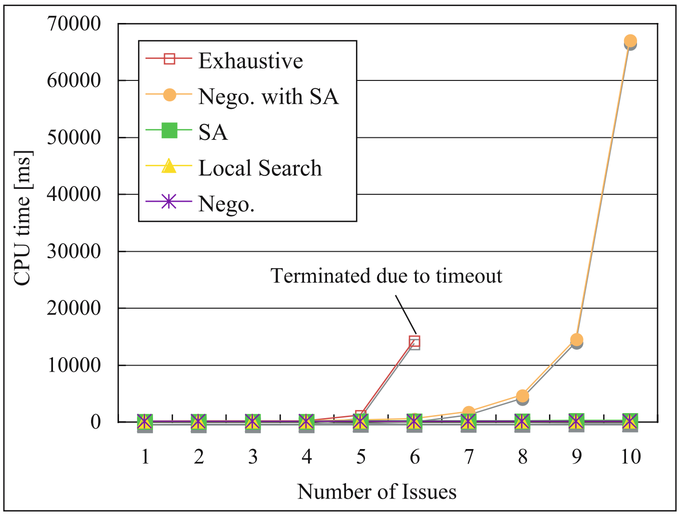

Fig. 9. Result 3: CPU time

Figure 9 shows the computational efficiency for each method. The exhaustive search is obviously worst. Simulated annealing search, local search-based negotiation and the negotiation without SA-sampling are computationally efficient. In terms of the negotiation with SA-sampling, when the number of issues is larger than 9, it needs a significant computational cost. This is because the number of bids are drastically increased in the parameters we set in this experiment. We are considering the two future directions. One is to tune the parameters. For example, if we set the threshold for identifying bids to a value that depends on the number of issues, it may have good results in terms of the computational cost. The other is to add the algorithm that can select more feasible bids.

5 Related Work

There are a lot of previous work on multi-issue negotiation [1–6]. These efforts differ from our work since our protocol is attacking against handle completely nonlinear utilities. We can find several previous efforts focus on nonlinear utilities.

Klein et al. [8] proposed an agent negotiation method for nonlinear utility models. A mediator agent effectively manages negotiation between two agents so that they reach a Pareto optimal agreement point. Our work originally inspired by this work. The difference is that we employ auction style method so that two or more agents can participate in our negotiation model.

Ito et al. [9] proposed a simple negotiation method for multi-issue negotiation and extend it for nonlinear utility domain. The protocol is based on a combinatorial auction protocol. However, it did not show sufficient result on nonlinear utility domain.

Lin et al. [10] proposed bilateral multi-issue negotiations for nonlinear utility models. They explored a range of protocols based on mutation and selection with binary contracts. (1) Multiple text proposal exchange: Each agent maintains a population of contracts, and proposes several of them at once, optionally annotated with that agent's preference information. At each step, one updates one's own population by selecting from the result of recombining the other agents' counter proposals with one's current population. Each agent keeps trying to increase contract utility, so it is a multiple negotiation text protocol rather a concession protocol. (2) Mediated multiple text negotiation: a mediator starts by generating a random set of possible contracts. Each agent identifies the subset it prefers. These subsets are recombined and mutated, forming a new set of candidates that the agents selects from. At some point, the agents rank order their preferred subsets, and the highest match represents the final agreement. The paper does not describe what kind of utility functions are used, nor does it present any experimental analyses. It is therefore unclear whether this strategy enables sufficient exploration of the strategy space to find win-win solutions with multi-optimal utility functions. But the idea does seem interesting.

The followings efforts focus on linear utility models.

Fatima et al. [11] proposed an agenda-based framework for multi-issue negotiation. They discussed mainly how to decide the order that issues should be negotiated in, which impacts efficiency and fairness. Issues are independent. The difference is that we employ auction methods and discuss the extension to nonlinear utility cases.

Jonker et al. [12–14] propose an agent architecture for multi-issue negotiation. However, they use a linear utility (weighted sum) model.

Luo et al. [15] proposed that proposal exchange approach wherein tradeoffs as well as concessions are used to seek a Pareto-optimal solution. Contracts are represented using (gradually tightening) fuzzy constraints so they represent a subspace rather than a single point. They model negotiation as a distributed constraint optimization problem with self-interested agents. Agents exchange proposals, relaxing their constraints over time, until there is an agreement. Preferences are modeled as prioritized fuzzy constraints (over one or more issues) are so they can be partially satisfied. Since they do allow one to express preferences over multiple attributes (e.g., cheap and distant is preferred over expensive and close) this does produce a multi-optimum utility function. They claim their algorithm is provably optimal, but do not discuss computational complexity and provide only a single small-scale example. The main difference is that we model multiple issues negotiation as generalized CSP, and assume competitive agents.

In Barbuceanu and Lo [16], a contract is defined as a goal tree, with a set of on/off labels for each goal (this defines the contract). A goal may represent, for example, the desire that an attribute value be within a given range. There are constraints that describe what patterns of on/off labels are allowable, as well as utility functions that describe, for each agent, what the utility of a given goal tree labeling is. This is essentially a binary-valued contract, except that the goal tree structure imposes some additional internal consistency constraints on what goals can be on or off (e.g., if a goal is on, so are all of its' subgoals; also, for disjunctive branches, only one of the subgoals can be on at a time). The total utility of a contract (they call it a set of on/off goal labels) is the sum of the utilities for each goal. They use a constraint solver algorithm to find the contracts that maximize the goal utilities plus satisfy as many constraints as possible, producing a multiple optimal utility function. It appears that all constraints are viewed as equally important. They claimed that their method is scalable. But very small example is shown and no theoretical analysis was shown. The main difference is that we employ auction method for resolving conflicts among competitive agents.

In Ito and Shintani [17, 18], a persuasion protocol was proposed. In the paper, people's preferences over multiple issues are quantified as a weighted hierarchy, using the Analytic Hierarchy Process (AHP). The weighted hierarchy involves problem issues and solution candidates. Each issue and solution candidate has a weighted values. In addition, by utilizing human's fuzzy weights, a software agent can change its preference when another agent persuades it to. Agents are not totally competitive in this study.

Distributed constraint satisfaction problem (DisCSP) [19] is a constraint satisfaction problem with distributed agents. DisCSP has not been assuming that agents are cooperative or competitive. However, in the DisCSP literature, the main results assume agents are cooperative [20, 21]. The difference is that we assume a generalized CSP among competitive agents, and give a negotiation protocol for that situation.

6 Conclusions and Future Work

Multi-issue negotiation protocols have been studied very widely. However, there have been very few work that focus on nonlinear utility spaces. In this paper, we assumed agents have nonlinear utility spaces. We proposed an auction-based multiple-issue negotiation protocol among nonlinear utility agents. Our negotiation protocol employs several techniques, i.e., adjusting sampling,auction-based maximization of social welfare. Our experimental results show that our method can outperform the existing simple methods in particular in the huge utility space that can be often found in the real world. Further, theoretically, our negotiation protocol can guarantee the completeness if some conditions are satisfied.

Interestingly, the exhaustive search often fails and cannot terminate if the utility space becomes huge. Also, when the utility space becomes huge and the number of constraints is not large, then the simulated annealing search often drop into local optimal. Even such cases our proposed method, the negotiation method with SA-sampling, can find approximately optimal points (we can not validate the points are optimal because the exhaustive search does not work in such a huge utility space).

In terms of future work, we push to scale up our method. If we increase the threshold for identifying bids, this reduces the number of bids and thus the winner determination computational cost decreases. We may also be able to take fewer samples, with hotter annealing at each sample point, since we expect fewer peaks if the threshold is high. However, increasing the threshold increases the risk of non-optimal outcomes since peaks that would belong to a Pareto-optimal negotiation outcome may be missed. So there is a computational cost/optimality tradeoff to be explored, which is affected by sampling rate, annealing temperature, and bid threshold. The next step is to clarify this tradeoff by tuning and sophisticating the negotiation method.

References

1. Wang, L.M., Huang, H.K., Chai, Y.M.: A learning-based multistage negotiation model. In: In Proceedings of International Conference on Machine Learning and Cybernetics. (2004) 140–145
2. Zhang, N., Zhang, S., Wang, L., Yang, J., Xu, Z.: Offer group generation and delayed processing in multi-issue negotiation. In: In Proceedings of IEEE/ WIC/ACM International Conference on Web Intelligence (WI 2004). (2004) 702–705
3. Rocha, A.P., Oliveira, E.: Adaptive multi-issue negotiation protocol for electronic commerce. In: In Proceedings of the Fifth International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 2000). (2000)
4. Soh, L.K., Li, X.: Adaptive, confidence-based multiagent negotiation strategy. In: In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS2004). (2004)
5. Fatima, S., Wooldridge, M., Jennings, N.R.: Optimal negotiation of multiple issues in incomplete information settings. In: In Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS2004). (2004)

6. Lau, R.Y.K.: Towards genetically optimised multi-agent multi-issue negotiations. In: In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS05). (2005)
7. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Englewood Cliffs, NJ (2002)
8. Klein, M., Faratin, P., Sayama, H., Bar-Yam, Y.: Negotiating complex contracts. Group Decision and Negotiation **12**(2) (2003) 58–73
9. Ito, T., Klien, M.: A multi-issue negotiation protocol among competitive agents and its extension to a nonlinear utility negotiation protocol. In: In Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS06). (2006) (to appear)
10. Lin, R.J., Chou, S.T.: Bilateral multi-issue negotiations in a dynamic environment. In: In Proceedings of the AAMAS Workshop on Agent Mediated Electronic Commerce (AMEC V). (2003)
11. Fatima, S.S., Wooldridge, M., Jennings, N.R.: An agenda-based framework for multi-issue negotiation. Artificial Intelligence **152** (2004) 1–45
12. Jonker, C.M., Jan T.: An agent architecture for multi-attribute negotiation. In: In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI2001). (2001) 1195–1202
13. Jonker, C.M., Robu, V.: Automated multi-attribute negotiation with efficient use of incomplete prefenrece information. In: In Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2004). (2004)
14. Tibor Bosse, C.M.J.: Human vs. computer behaviour in multi-issue negotiation. In: In Proceedings of 1st International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS2005). (2005)
15. Luo, X., Jennings, N.R., Shadbolt, N., fung Leung, H., man Lee, J.H.: A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. Artificial Intelligence **148** (2003) 53–102
16. Barbuceanu, M., Lo, W.K.: Multi-attribute utility theoretic negotiation for electronic commerce. In: In Proceedings of the International Workshop on Agent-mediated Electronic Commerce (AMEC2000). (2000)
17. Ito, T., Shintani, T.: Persuasion among agents: An approach to implementing a group decision support system based on multi-agent negotiation. In: In Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI97). (1997) 592–597
18. Shintani, T., Ito, T.: An architecture for multi-agent negotiation using private preferences in a meeting scheduler. In: In Proceedings of the 5th Pacific Rim International Conferences on Artificial Intelligence (PRICAI'98) (Lecture Notes in Artificial Intelligence 1531, PRICAI'98: Topics in Artificial Intelligence, Hing-Yan Lee and Hiroshi Motoda (Eds.), Springer). (1998) 47–58
19. Yokoo, M.: Distributed Constraint Satisfaction. Springer, Berlin Heidelberg New York (2001)
20. Yokoo, M., Sakurai, Y., Matsubara, S.: Robust combinatorial auction protocol against false-name bids. Artificial Intelligence **130**(2) (2001) 167–181
21. Modi, P.J., Shen, W.M., Tambe, M., Yokoo, M.: Adopt: Asynchronous distributed constraint optimization with quality guarantees. Artificial Intelligence **161** (2005) 149–180